Three-dimensional Taylor Green

1 Introduction

The Taylor-Green vortex at a Reynolds number of 1600 is a configuration commonly studied flow that captures elements of rudimental turbulence generation and decay [1]. This canonical case serves as a testbed for a high-order workshop [2] where various schemes can be compared and contrasted. Unstructured, low-Mach numerical approaches were presented in [3] using a series of homogenous topologies consisting of the following: Hex8, Hex27, Tet4, Pyramid5, and Wedge6 where convergence and relative accuracy was assessed relative to the reference direct numerical simulation (DNS).

The Taylor-Green flow regime is marked by three phases that are described as the following: Phase 1, viscous effects with small-scale laminar and organized structures are found; Phase 2, viscous (diffusion) effects dominating with accompanying stretching of vortex lines, and Phase 3, a break-up and is nearly isotropic in nature. Generally, for this simulation study, an explicit LES sub-grid stress model is omitted.

The initial condition for the three-dimensional flow field is as follows:

$$\begin{aligned} u_x &= u_o sin(\frac{x}{L})cos(\frac{y}{L})cos(\frac{z}{L}), \\ u_y &= -u_o cos(\frac{x}{L})sin(\frac{y}{L})cos(\frac{z}{L}), \\ u_z &= 0, \\ p &= p_o + \frac{\rho_o u_o^2}{16}\left(cos(\frac{2x}{L}) + cos(\frac{2y}{L})\right)\left(cos(\frac{2z}{L}) + 2\right). \end{aligned} \tag{1}$$

The quantities of interest (QoIs) for this simulation study are the temporal evolution of the kinetic energy, E_k, that is integrated over the full domain at each time step, the integrated dissipation rate as a function of time, and the temporal enstrophy response. The energy-based dissipation rate is given by $\epsilon_1 = -\frac{dE_k}{dt}$. The integrated enstrophy, ζ is defined by (in a low-Mach flow), $\epsilon_2 = \frac{2\mu}{\rho_o}\zeta$, where,

$$\zeta = \frac{1}{\rho_o V}\int_\Omega \frac{1}{2}\rho\omega_k\omega_k d\Omega. \tag{2}$$

For more details, see [4].

2 Domain

The simulation is run in a periodic square box of $-\pi L \leq x, y, z \leq \pi L$ with L equal to unity. The simulation is allowed to evolve from this initial condition over 20 characteristic convection time scales (defined by $t_c = \frac{L}{u_o}$). For this configuration, peak dissipation occurs at approximately $8t_c$.

2.1 Simulation Specification and Sample Results

A sample Q-criterion set of images for a Taylor-Green Re 1600 image appearing in [3] is shown in Figure 1, while the set of converged results in this same paper are shown in Figure 2.

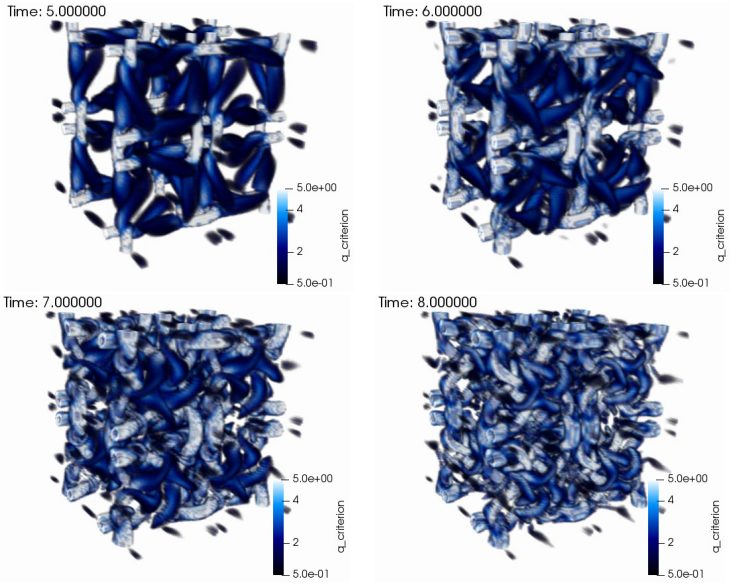

Figure 1: Three-dimensional Q-criterion volume rendered Taylor-Green representative time series at 5, 6, 7, and 8 seconds.

2.2 Meshes

The set of meshes provided in the /mesh directory are as follows:

- 3d_hex8_taylor_green_0p2.g (Hex8 topology using 0.2 mesh spacing)
- 3d_tet4_taylor_green_0p2.g (Tet4 topology using 0.2 mesh spacing)
- 3d_tet4_taylor_green_0p4.g (Tet4 topology using 0.4 mesh spacing)

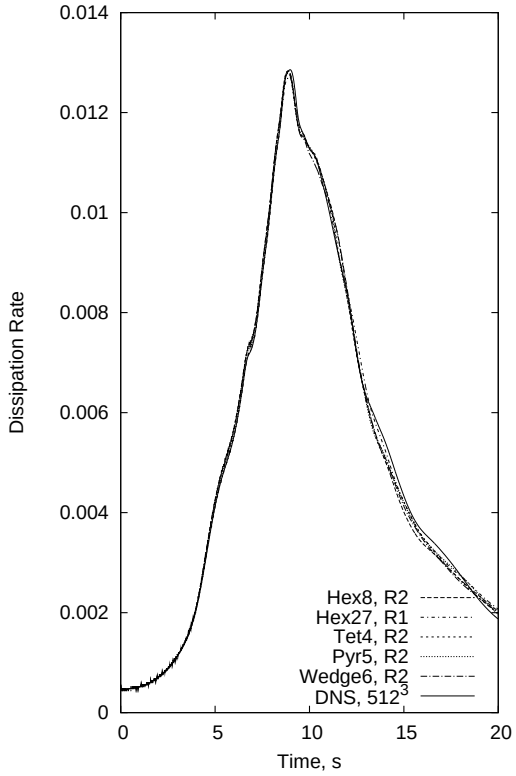

Figure 2: Turbulence dissipation rate temporal plot for the series of mesh refinements in [3]. In this study, the baseline (R0) Hex8 mesh spacing of $\frac{2\pi}{100L}$, i.e., 100^3 elements was used.

The "0p4" mesh can be useful for trouble shooting the monolithic code implementation. Note that most peer-reviewed results appearing in the open literature exercise meshes that, even for a coarsest mesh spacing reported, are far more refined than the meshes provided in this study. However, the prime motivation for this exercise is to learn more about fully implicit solver approaches in the low-Mach limit.

3 Equation Set

The variable-density low-Mach equation set is defined by the continuity and momentum equation, here shown in integral form,

$$\int \frac{\partial \rho}{\partial t} dV + \int \rho \hat{u}_j n_j dS = 0, \tag{3}$$

$$\int \frac{\partial \rho u_i}{\partial t} dV + \int \rho \hat{u}_j u_i n_j dS - \int 2\mu S^*_{ij} n_j dS = -\int P \delta_{ij} n_j dS \tag{4}$$

In the above equation, ρ is the fluid density and u_i is the fluid velocity, while the traceless rate-of-strain tensor is defined as

$$S^*_{ij} = S_{ij} - \frac{1}{3}\delta_{ij} S_{kk} = S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}.$$

In a low-Mach flow, the above pressure, P, is the perturbation about the thermodynamic pressure, P^{th}. We also note that the form shown above includes pressure stabilization embedded in the special velocity, $\hat{u}_j$, whose form can be written as,

$$\rho \hat{u}_j = \rho u_j - \frac{\Delta t}{\gamma_1}\left(\frac{\partial P}{\partial x_j} - G_j P\right), \tag{5}$$

where the projected nodal gradient, $G_j P$ is obtained though the following equation:

$$\int G_j P dV - \int P n_j dS = 0. \tag{6}$$

When using higher-order approaches, and to retain design-order accuracy, $G_j P$ must be computed using the consistent mass matrix, while using mass-lumping, yields,

$$G_j P = \frac{\int P n_j dS}{\int dV}. \tag{7}$$

As captured in the tutorial notes, this Rhie-Chow-like form [5] can be shown to be an algebraic manipulation of the assembled momentum system to approximate the fine-scale momentum equation (see also [6]). In general, the states associated with the projected nodal gradient can drastically improve nonlinear convergence as can the time scale chosen (above, the simulation time step, Δt is used). In most cases, the projected nodal gradient is lagged by one iteration since inclusion of this equation can increase the implicitly coupled system. As such, quadratic nonlinear convergence within a time step can be affected when using this lagged approach since sensitivities to the fully coupled system are missing. However, choosing this value to be the state n quantity can improve convergence, while influencing time accuracy.

There are also instances where the effect of pressure stabilization is not propagated to the momentum equation and, if in use, any other transport equation, e.g., mass fraction, enthalpy, etc. In the simplest form, $\rho\hat{u}_j = \rho u_j$, while in Charnyi et al. [7], the termed energy, momentum and angular momentum (EMA) method is described.

4 Discussion Points

There are several interesting activities associated with this sample case including the points captured below.

- Document the appropriate equations for this conceptual model problem. Feel free to enforce the incompressible constraint since this approach simplifies the continuity equation and both the momentum time and viscous term.

- Run the pressure projection input file (modified to run at least 20 seconds), while post-processing the integrated kinetic energy and dissipation plots. For the reference DNS of [2], see the "/data" directory. In recent versions of Nalu, the log file for the simulation will contain values such as "sum(vol), sum(ke*vol), mean(ke)" that represents the total volume, integrated kinetic energy, and mean kinetic energy of the system at each converged time step. Use both the 0p2 Hex8 and Tet4 mesh in this study, while visualizing the flow field. Comment on the kinetic energy and dissipation rate time history for each topology relative to each other and the DNS. Note that you will need to differentiate the kinetic energy data in time to obtain the dissipation rate plot. Most plotting packages, e.g., Matlab, Tecplot, etc., support this post-processing step.

- Implement an interior monolithic momentum and continuity kernel that captures the equation contributions identified in the above first step. In fact, this kernel should look very much like a combination of the segregated continuity (ContinuityAdvElemKernel) and momentum (MomentumAdvDiffElemKernel and MomentumMassElemKernel) classes, while noting that the total size of the new monolithic system represented in your new

kernel is now $nDim + 1$. This differs from the segregated kernels where momentum is size $nDim$ and continuity is unity in size. In the LowMachMonolithicEquationSystem class, a single kernel contribution is expected by the name of:

  - uvwp_time_advection_diffusion (a consistent-mass kernel) or,
  - uvwp_lumped_time_advection_diffusion (a lumped-mass kernel)

  A purely diagonal monolithic system can be obtained simply by copying the code from the segregated Continuity and Momentum kernels and by modifying the total size of the matrix system. As a hint, the easiest approach may be to use the 0, 1, and 2 memory location for the momentum contribution, and the "AlgTraits::nDim_" entry for continuity. Include the code base in the report prepared as an appendix item. Of coarse, the objective of this laboratory is to include the cross Jacobian entries, e.g., $\frac{\partial C}{\partial u_j}$ and $\frac{\partial U_j}{\partial P}$, where C and U_j represent the continuity and momentum equation. The 0p4 Tet4 mesh may be useful for debugging.

- Run the monolithic kernel that you have implemented and compare nonlinear convergence between the production pressure projection scheme and your monolithic kernel. Comment on overall speed to reach the end simulation time.

- Explore the usage of pressure stabilization in the momentum mass flow rate expression, in addition to using the "old" values for the momentum and projected nodal pressure gradient field. You may also explore other forms of the advection form, i.e., EMA, although do not worry about implementing any upwind or residual-based operator to manage advection stabilization.

- Compare and contrast the simulation timings between the pressure projection and monolithic implementation at varying nonlinear tolernaces.

- $Optional, not-required$: As time permits, perform any mesh refinement and comment on the sensitivity of the QoIs.

- $Optional, not-required$: Perform any appropriate code verification using, for example, the convecting Taylor Vortex analytical solution on a periodic domain using your newly coded monolithic implementation.

- $Optional, not-required$: Explore other forms of the advection operator, i.e., EMA and report findings as compared to the advection form shown in this report.

- $Optional, not-required$: Implement various upwind approaches via the usage of the nodal field "dudx".

- $Optional, not-required$: Implement a residual-based stabilization approach and comment on how this usage affects both convergence and QoIs.

- *Optional*, $not - required$: Perform mesh refinement and capture results compared to the DNS reference solution.

References

[1] G. I. Taylor and A. E. Green. Mechanism of the production of small eddies from large ones. *Proc. Roy. Soc. Lond.*, 158:499–521, 1937.

[2] K. Hillewaert. First international workshop on high-order cfd methods. Number AIAA 2012 in 50th Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meeting, Nashville, TN, Jan. 7–8, 2012, AIAA, Washington, D.C., 2012. AIAA.

[3] S. Domino, P. Sakievich, and M. Barone. An assessment of atypical mesh topologies for low-Mach large-eddy simulation. *Comp. Fluids*, 179:655–669, 2019.

[4] J. R. Bull and A. Jameson. Simulation of the taylor–green vortex using high-order flux reconstruction schemes. *AIAA Journal*, 53(9):2750–2761, 2015.

[5] C. M. Rhie and W. L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21(11):1525–1532, 1983.

[6] J. Martínez, F. Piscaglia, A. Montorfano, A. Onorati, and S.M. Aithal. Influence of momentum interpolation methods on the accuracy and convergence of pressure–velocity coupling algorithms in openfoam®. *Journal of Computational and Applied Mathematics*, 309:654–673, 2017.

[7] Sergey Charnyi, Timo Heister, Maxim A. Olshanskii, and Leo G. Rebholz. On conservation laws of navier–stokes galerkin discretizations. *Journal of Computational Physics*, 337:289–308, 2017.